AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listing of claims in the application:

Claim 1 (currently amended): A method of simulating a circuit having a hierarchical data structure, comprising:

representing the circuit as a hierarchically arranged set of branches, including a root branch and a plurality of other branches logically organized in a graph; the hierarchically arranged set of branches including a first branch that includes one or more leaf circuits and a second branch that includes one or more leaf circuits; wherein the first branch and second branch are interconnected in the graph through a third branch at a higher hierarchical level in the graph than the first and second branches;

creating a static database in accordance with a netlist description of the circuit, the static database including topology information of the circuit;

selecting a group circuit for simulation, the group circuit comprises one or more leaf circuits selected from the first branch and the second branch;

creating a dynamic database for representing the group circuit, the dynamic database including references to corresponding branches of the hierarchical data structure in the static database for fetching topology information dynamically during simulation; and

simulating the group circuit in accordance with the dynamic database.

Claim 2 (original): The method of claim 1, wherein the step of creating a static database comprises:

partitioning the circuit into a hierarchical data structure consisting one or more static branch circuits, each static branch circuit containing one or more functional calls, wherein each functional call contains reference to another static branch circuit or to a static leaf circuits at a lower hierarchy; and

identifying topology information between the static branch circuits and the static leaf circuits.

Claim 3 (original): The method of claim 2, wherein the step of identifying topology information further comprises:

determining whether a topological graph of two or more circuits are substantially the same; and

creating a hierarchical data structure with references to one common circuit for representing the two or more substantially the same circuits.

Claim 4 (original): The method of claim 1, wherein the step of creating the static database further comprises:

flattening a selected group of leaf circuits to form a single flatten leaf circuit;

representing resistor-capacitor networks with corresponding electrically substantially equivalent resistor-capacitor networks having fewer resistor or capacitor elements; and

combining tightly coupled leaf circuits into a single merged leaf circuit in accordance with a set of predefined coupling conditions.

Claim 5 (original): The method of claim 1, wherein the step of creating the dynamic database comprises:

creating a group matrix solver for solving the matrix representation of the one or more leaf circuits within the group circuit;

creating one or more input ports for each leaf circuit in the group;

creating one or more output ports for each leaf circuit in the group;

creating one or more loads for each leaf circuit in the group, wherein each load represents the impedance and capacitance observed at the input port of a leaf circuit; and

creating a port connectivity interface for connecting the input ports, output ports and loads between the leaf circuits, wherein the port connectivity interface communicates changes in signal conditions among the one or more leaf circuits within the group circuit.

Claim 6 (original): The method of claim 5, wherein the port connectivity interface comprises:

a set of input vectors for referencing to a set of input ports of one or more receiver leaf circuits;

a set of output vectors for referencing to a set of output ports of one or more driver leaf circuits;

a set of load vectors for referencing to a set of loads of the one or more driver leaf circuits; and

an array of storage elements for storing information associating the set of loads to the set of input ports.

Claim 7 (original): The method of claim 1, wherein the step of creating the dynamic database further comprises:

creating one or more dynamic branch circuits mirroring the static branch circuits in the hierarchical data structure, each dynamic branch circuit containing one or more functional calls, each functional call contains reference to another dynamic branch circuit or to a dynamic leaf circuit at a lower hierarchy; and

identifying connectivity information between the dynamic branch circuits and the dynamic leaf circuits.

Claim 8 (currently amended): A system for simulating a circuit having a hierarchical data structure, comprising:

at least one processing unit for executing computer programs;

a user interface for performing at least one of the functions selected from the group consisting of entering a netlist representation of the circuit, viewing representations of the circuit on a display, and observing simulation results of the circuit;

a memory for storing a static database and a dynamic database of the circuit;

means for representing the circuit as a hierarchically arranged set of branches, including a root branch and a plurality of other branches logically organized in a graph; the hierarchically arranged set of branches including a first branch that includes one or more leaf circuits and a second branch that includes one or more leaf circuits; wherein the first branch and second branch are

interconnected in the graph through a third branch at a higher hierarchical level in the graph than the first and second branches;

means for creating a static database in accordance with a netlist description of the circuit, the static database including topology information of the circuit;

means for selecting a group circuit for simulation, the group circuit comprises one or more leaf circuits selected from the first branch and the second branch;

means for creating a dynamic database for representing the group circuit, the dynamic database including references to corresponding branches of the hierarchical data structure in the static database for fetching topology information dynamically during simulation; and

means for simulating the group circuit in accordance with the dynamic database.

Claim 9 (original): The system of claim 8, wherein the means for creating a static database comprises:

means for partitioning the circuit into a hierarchical data structure consisting one or more static branch circuits, each static branch circuit containing one or more functional calls, wherein each functional call contains reference to another static branch circuit or to a static leaf circuits at a lower hierarchy; and

means for identifying topology information between the static branch circuits and the static leaf circuits.

Claim 10 (original): The system of claim 9, wherein the means for identifying topology information further comprises:

means for determining whether a topological graph of two or more circuits are substantially the same; and

means for creating a hierarchical data structure with references to one common circuit for representing the two or more substantially the same circuits.

Claim 11 (original): The system of claim 8, wherein the means for creating the static database further comprises:

means for flattening a selected group of leaf circuits to form a single flatten leaf circuit;

means for representing resistor-capacitor networks with corresponding electrically substantially equivalent resistor-capacitor networks having fewer resistor or capacitor elements; and

means for combining tightly coupled leaf circuits into a single merged leaf circuit in accordance with a set of predefined coupling conditions.

Claim 12 (original): The system of claim 8, wherein the means for creating the dynamic database comprises:

means for creating a group matrix solver for solving the matrix representation of the one or more leaf circuits within the group circuit;

means for creating one or more input ports for each leaf circuit in the group;

means for creating one or more output ports for each leaf circuit in the group;

means for creating one or more loads for each leaf circuit in the group, wherein each load represents the impedance and capacitance observed at the input port of a leaf circuit; and

means for creating a port connectivity interface for connecting the input ports, output ports and loads between the leaf circuits, wherein the port connectivity interface communicates changes in signal conditions among the one or more leaf circuits within the group circuit.

Claim 13 (original): The system of claim 12, wherein the port connectivity interface comprises:

a set of input vectors for referencing to a set of input ports of one or more receiver leaf circuits;

a set of output vectors for referencing to a set of output ports of one or more driver leaf circuits;

a set of load vectors for referencing to a set of loads of the one or more driver leaf circuits; and

an array of storage elements for storing information associating the set of loads to the set of input ports.

Claim 14 (original): The system of claim 8, wherein the means for creating the dynamic database further comprises:

means for creating one or more dynamic branch circuits mirroring the static branch circuits in the hierarchical data structure, each dynamic branch circuit containing one or more functional calls, each functional call contains reference to another dynamic branch circuit or to a dynamic leaf circuit at a lower hierarchy; and

means for identifying connectivity information between the dynamic branch circuits and the dynamic leaf circuits.

Claim 15 (currently amended): A computer program product, comprising a medium storing computer programs for execution by one or more computer systems, the computer program product comprising:

a simulator module for simulating a circuit having a hierarchical data structure, wherein the simulator module is used in conjunction with at least a processing unit, a user interface and a memory, and the simulator module includes one or more computer programs containing instructions for:

representing the circuit as a hierarchically arranged set of branches, including a root branch and a plurality of other branches logically organized in a graph; the hierarchically arranged set of branches including a first branch that includes one or more leaf circuits and a second branch that includes one or more leaf circuits; wherein the first branch and second branch are interconnected in the graph through a third branch at a higher hierarchical level in the graph than the first and second branches;

creating a static database in accordance with a netlist description of the circuit, the static database including topology information of the circuit;

selecting a group circuit for simulation, the group circuit comprises one or more leaf circuits selected from the first branch and the second branch;

creating a dynamic database for representing the group circuit, the dynamic database including references to corresponding branches of the hierarchical data structure in the static database for fetching topology information dynamically during simulation; and

simulating the group circuit in accordance with the dynamic database.

Claim 16 (original): The computer program product of claim 15, wherein the instructions for creating a static database comprise:

partitioning the circuit into a hierarchical data structure consisting one or more static branch circuits, each static branch circuit containing one or more functional calls, wherein each functional call contains reference to another static branch circuit or to a static leaf circuits at a lower hierarchy; and

identifying topology information between the static branch circuits and the static leaf circuits.

Claim 17 (original): The computer program product of claim 16, wherein the instruction for identifying topology information further comprises:

determining whether a topological graph of two or more circuits are substantially the same; and

creating a hierarchical data structure with references to one common circuit for representing the two or more substantially the same circuits.

Claim 18 (original): The computer program product of claim 15, wherein the instructions for creating the static database further comprise:

flattening a selected group of leaf circuits to form a single flatten leaf circuit;

representing resistor-capacitor networks with corresponding electrically substantially equivalent resistor-capacitor networks having fewer resistor or capacitor elements; and

combining tightly coupled leaf circuits into a single merged leaf circuit in accordance with a set of predefined coupling conditions.

Claim 19 (original): The computer program product of claim 15, wherein the instructions for creating the dynamic database comprise:

creating a group matrix solver for solving the matrix representation of the one or more leaf circuits within the group circuit;

creating one or more input ports for each leaf circuit in the group;

creating one or more output ports for each leaf circuit in the group;

creating one or more loads for each leaf circuit in the group, wherein each load represents the impedance and capacitance observed at the input port of a leaf circuit; and

creating a port connectivity interface for connecting the input ports, output ports and loads between the leaf circuits, wherein the port connectivity interface communicates changes in signal conditions among the one or more leaf circuits within the group circuit.

Claim 20 (original): The computer program product of claim 19, wherein the port connectivity interface comprises:

a set of input vectors for referencing to a set of input ports of one or more receiver leaf circuits;

a set of output vectors for referencing to a set of output ports of one or more driver leaf circuits;

a set of load vectors for referencing to a set of loads of the one or more driver leaf circuits; and

an array of storage elements for storing information associating the set of loads to the set of input ports.

Claim 21 (original): The computer program product of claim 15, wherein the instructions for creating the dynamic database further comprise:

creating one or more dynamic branch circuits mirroring the static branch circuits in the hierarchical data structure, each dynamic branch circuit containing one or more functional calls, each functional call contains reference to another dynamic branch circuit or to a dynamic leaf circuit at a lower hierarchy; and

identifying connectivity information between the dynamic branch circuits and the dynamic leaf circuits.